

CLAIMS:

1. A method of generating executable program code for a data processing system, the method comprising:
 - 5 an encoding stage for generating a compressed intermediate representation (E-IR) of an input code (IC), the encoding stage comprising:
 - transforming (301) the input code including performing a selected set of code optimisation steps resulting in transformed code (302) and compiler information (303) about the transformed code;
 - extracting (304,306) state information (305) of a statistical model and statistical information (PDF) from the transformed code and the compiler information; and
 - encoding (307) the transformed code and the compiler information using the extracted state information and statistical information and resulting in the compressed intermediate representation;
 - 10 a decoding stage for generating the executable program code from the compressed intermediate representation, the decoding stage comprising:
 - decoding (401) the compressed intermediate representation resulting in the transformed code (402) and the compiler information (403); and
 - further compiling (407) the transformed code using the decoded compiler information and resulting in the executable program code (EXE).
 - 15
- 20 2. A method according to claim 1, characterised in that the encoding stage is performed on a first data processing system (101) and the decoding stage is performed on a second data processing system (102); the method further comprising transferring the compressed intermediate representation from the first data processing system to the second data processing system.
- 25 30 3. A method according to claim 1 or 2, characterised in that the step of generating the state information and statistical information further comprises

- obtaining state information (305) from a state machine (304) based on the transformed code and the compiler information; and
- obtaining probability information (PDF) from a statistical model (306) based on the obtained state information.

5 4. A method according to claim 3, characterised in that the state machine comprises a syntactic model of at least one of the transformed code and the compiler information.

10 5. A method according to claim 3 or 4, characterised in that the state machine comprises an execution model of the transformed code.

6. A method according to any one of claims 3 through 5, characterised in that the state machine comprises a model of the compiler information.

15 7. A method according to any one of claims 1 through 6, characterised in that the method further comprises

- storing the compressed intermediate representation; and
- performing the decoding stage in connection with a subsequent execution

20 20 of the generated executable program code.

8. A method according to any one of claims 1 through 7, characterised in that the step of further compiling the transformed code further comprises further optimising the resulting executable code.

25 9. A method according to any one of claims 1 through 8, characterised in that the input code comprises Java bytecode.

30 10. A method according to any one of claims 1 through 9, characterised in that the data processing system is a mobile terminal.

11. A method according to any one of claims 1 through 10, characterised in that the transformed code comprises a number of code elements and the

method further comprises determining a probability distribution of said code elements and providing the determined probability distribution to the step of generating statistical information.

- 5 12. A method of generating program code for a data processing system, the method comprising
 - transforming (301) an input code (IC) including performing a selected set of optimisation steps resulting in transformed code (302) and compiler information (303) about the transformed code;
 - extracting (304,306) state information (305) of a statistical model and statistical information (PDF) from the transformed code and the compiler information; and
 - encoding (307) the transformed code and the compiler information using the extracted state information and statistical information and resulting in a compressed intermediate representation (E-IR) of the input code; the compressed intermediate representation being adapted to be decoded and further compiled in a subsequent decoding stage for generating executable program code.
- 10 13. A computer program comprising program code means for performing all the steps of claim 12 when said program is run on a computer.
- 15 14. A method of generating executable program code in a data processing system, the method comprising
 - receiving a compressed intermediate representation (E-IR) of an input code, the compressed intermediate representation including encoded transformed code generated and at least partially optimised by a compiler and encoded compiler information indicative of further information generated by the compiler about the transformed code, the encoded transformed code and the encoded compiler information being encoded using state information of a statistical model and statistical information extracted from the transformed code and the compiler information;
- 20
- 25
- 30

- decoding (401) the compressed intermediate representation resulting in the transformed code (402) and the compiler information (403); and
- further compiling (407) the transformed code using the decoded compiler information and resulting in the executable program code (EXE).

5

15. A computer program comprising program code means for performing all the steps of claim 14 when said program is run on a computer.

10 16. A data processing system for generating executable program code, the system comprising

an encoding module adapted to generate a compressed intermediate representation of an input code, the encoding module comprising:

- first compiler means adapted to transform the input code including performing a selected set of code optimisation steps and resulting in transformed code and compiler information about the transformed code;
- processing means adapted to extract state information of a statistical model and statistical information from the transformed code and the compiler information; and
- 20 - encoding means adapted to encode the transformed code and the compiler information using the extracted state information and statistical information and resulting in the compressed intermediate representation;

25

a decoding module adapted to generate the executable program code from the compressed intermediate representation, the decoding module comprising:

- decoding means adapted to decode the compressed intermediate representation resulting in the transformed code and the compiler information; and
- 30 - second compiler means adapted to further compile the transformed code using the decoded compiler information and resulting in the executable program code.

17. An encoding device for generating program code for a data processing system, the encoding device comprising

- compiler means adapted to transform an input code including performing a selected set of code optimisation steps resulting in transformed code and compiler information about the transformed code;
- processing means adapted to extract state information of a statistical model and statistical information from the transformed code and the compiler information; and
- encoding means adapted to encode the transformed code and the compiler information using the extracted state information and statistical information and resulting in a compressed intermediate representation of the input code; the compressed intermediate representation being adapted to be decoded and further compiled in a subsequent decoding stage for generating executable program code.

15

18. A data processing system for generating executable program code, the data processing system comprising

- receiving means adapted to receive a compressed intermediate representation of an input code, the compressed intermediate representation including encoded transformed code generated and at least partially optimised by a compiler and encoded compiler information generated by the compiler indicative of further information generated by the compiler, the encoded transformed code and the encoded compiler information being encoded using state information of a statistical model and statistical information extracted from the transformed code and of the compiler information;
- decoding means adapted to decode the compressed intermediate representation resulting in the transformed code and the compiler information; and
- compiler means adapted to further compile the transformed code using the decoded compiler information and resulting in the executable program code.

19. A data record comprising a compressed intermediate representation of an input code, the compressed intermediate representation including encoded transformed code generated and at least partially optimised by a compiler and encoded compiler information indicative of further information
5 generated by the compiler, the encoded transformed code and the encoded compiler information being encoded using state information of a statistical model and statistical information extracted from the transformed code and from the compiler information; the compressed intermediate representation being adapted to be decoded and further compiled by a data processing system resulting in executable program code.
10